CHAPTER 5

Upgrading to Adaptive Server Anywhere

About this chapter

This chapter describes upgrade procedures for users of SQL Anywhere Versions 5.0 and 5.5, and users of Watcom SQL 4.0.

Contents

Understanding upgrades

A database application and DBMS consists of several components. If your organization has a large SQL Anywhere installation, you may have many client machines, possibly running several applications, connected to more than one server.

When upgrading your system, you need to decide which components to upgrade, and in which order. This chapter guides you in making those choices and carrying out the upgrade.

SQL Anywhere Version 5
In this chapter, **SQL Anywhere Version 5** refers to both versions 5.0 and 5.5 of SQL Anywhere.

☞ In order to upgrade existing applications to Adaptive Server Anywhere Version 6, it is helpful to understand how Version 6 connections work. For information, see "Connecting to a Database" on page 31 of the book *Adaptive Server Anywhere User's Guide*.

Standard upgrade precautions

There are several precautions you should take before upgrading any application, and these apply to Adaptive Server Anywhere upgrades just as to any other software.

- **Check the behavior changes** Before upgrading, you should confirm that none of the documented behavior changes in Version 6 affect your application.

  ☞ For information on behavior changes, see "Behavior Changes" on page 51.

- **Test** You should test your application in a Version 6 environment thoroughly before upgrading any applications in production use.
- **Backup** You should back up your existing software and data before upgrading.
- **Test your upgrade procedure** If you are upgrading many end users, test your upgrade procedure carefully before carrying it out.

SQL Anywhere is used in so many different configurations that no upgrade guidelines can be guaranteed for all cases.

The components in your system

If you are currently running SQL Anywhere Version 5, you will have some or all of the following components in your system:

- **Application** Your application, aside from the SQL Anywhere components.
- **Connection parameters** SQL Anywhere Version 5.0 connection parameters. These may be assembled from an ODBC data source, or in some other way.
- **Driver manager** The ODBC driver manager, for ODBC applications.
- **ODBC driver** The SQL Anywhere Version 5 ODBC driver, for ODBC applications. For network applications, the ODBC driver is on the client machine.
- **Interface library** The SQL Anywhere Version 5 interface library is used by ODBC and Embedded SQL applications. For network applications, the interface library is on the client machine.
- **SQL Anywhere client** The *dbclient.exe* executable and its command line, for network applications. The command line may specify the server name, and a set of network communications parameters. It may be stored in a batch file or an ODBC data source Start Line parameter. For network applications, the SQL Anywhere client is on the client machine.
- **The database server** The SQL Anywhere Version 5 database server. For network applications, this may be on a separate machine from the client components.
- **The database** A SQL Anywhere Version 5 database. This is on the same machine as the database server.

Database upgrades not required
You do not need to upgrade your database in order to use the Version 6 database server.

Major upgrading issues

The major issues in upgrading arise from the change in client/server communication protocol. This change means that the Version 5 interface library is not able to communicate with a Version 6 database server.

To help with this issue, Adaptive Server Anywhere Version 6 includes a **compatibility library**. This DLL allows communications to both Version 5 and Version 6 database servers.

☞ The new client/server communications protocol is described in "New communications features" on page 32.

When to upgrade your database

There is no need to upgrade the database itself when you start using Version 6. You need to upgrade the database only if you wish to take advantage of Version 6 features.

☞ For more information on upgrading databases, see "Upgrading databases" on page 86.

Running more than one version of the software

The Adaptive Server Anywhere software is designed so that both Version 5 and Version 6 can be run if you install Adaptive Server Anywhere Version 6 in a separate directory from SQL Anywhere Version 5. This is the default behavior of the Adaptive Server Anywhere installation program.

Review of SQL Anywhere Version 5 architecture

This section reviews the architecture for SQL Anywhere Version 5 applications connecting to a SQL Anywhere Version 5 database.

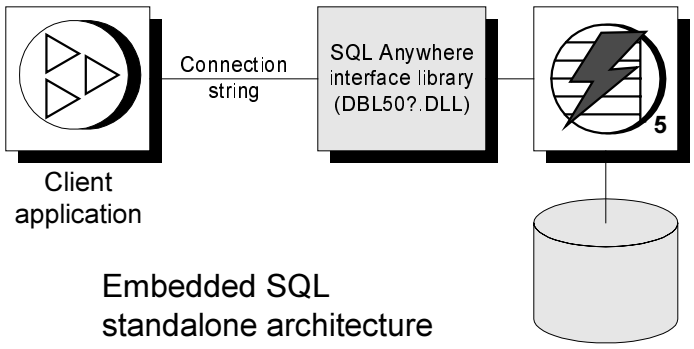
This information helps you to understand the changes needed when upgrading to Version 6. If you are familiar with SQL Anywhere Version 5 architecture, you do not need to read this section.

Standalone components for Version 5 (Embedded SQL)

If you are using SQL Anywhere Version 5 as a personal server, with an Embedded SQL client application, you are using the following components on your machine:

- A SQL Anywhere Version 5 database.
- The SQL Anywhere Version 5 database engine (personal database server).
- The SQL Anywhere Version 5 interface library.
- A SQL Anywhere Version 5.0 connection string.

The following figure illustrates how these pieces fit together.



Here, the question mark in *dbl50?.dll* represents a single character indicating the operating system. The interface library is named *dlb50t.dll* on Windows 95 and Windows NT, *dbl50w.dll* on Windows 3.x, and so on.

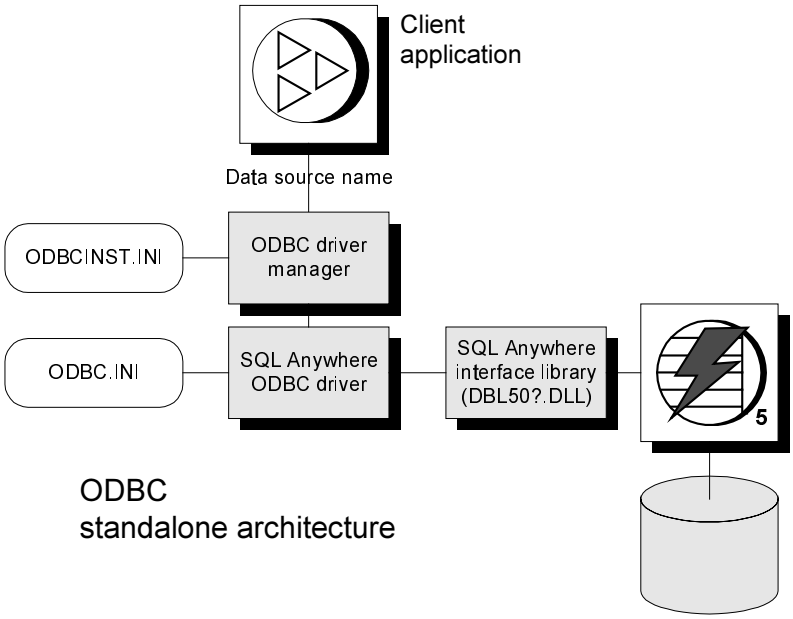
Embedded SQL client applications make calls to the interface library. The interface library is referenced by name, so that the name *dbl50?.dll* is a part of the client application.

Standalone components for Version 5 (ODBC)

If you are using SQL Anywhere Version 5 as a personal server, with an ODBC client application, you are using the following components on your machine:

- A SQL Anywhere Version 5 database.
- The SQL Anywhere Version 5 database engine.
- The SQL Anywhere Version 5 interface library.
- The SQL Anywhere Version 5 ODBC driver.
- The ODBC driver manager.
- A SQL Anywhere Version 5.0 connection description. This may be an ODBC data source, or a connection string from an application.

The following figure illustrates how these pieces fit together. The client application passes a data source name to the ODBC driver manager. The ODBC driver manager looks up the appropriate driver in ODBCINST.INI. The driver looks up the connection information in ODBC.INI and, via the interface library, connects to the SQL Anywhere Version 5 database engine.



ODBC standalone architecture

Client/server components for Version 5

If you are using SQL Anywhere as a network server, you have the following components on your server machine:

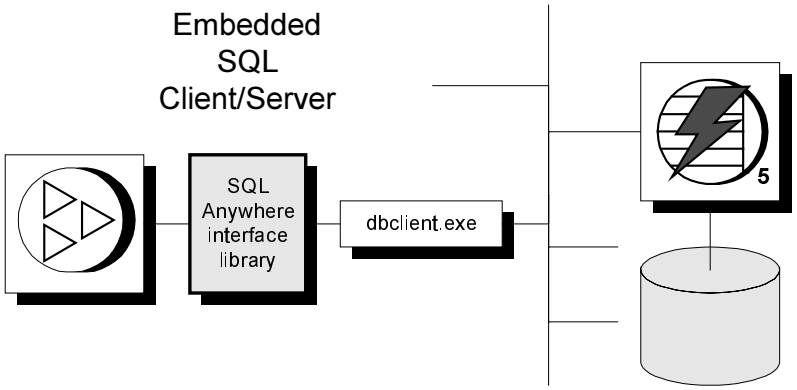
- A SQL Anywhere Version 5 database
- The SQL Anywhere Version 5 database server

You have the following components on your client machine:

- The SQL Anywhere Version 5 Client executable
- The SQL Anywhere Version 5 interface library.
- The SQL Anywhere ODBC driver, if your application uses ODBC.
- A SQL Anywhere Version 5.0 connection description. This may be an ODBC data source, or a connection string from an application.

  The data source may contain connection information in the start option, corresponding to a SQL Anywhere Client command line. You may also have connection strings in your application, and batch files that start a client with particular command-line parameters.

The architecture of a Version 5 Embedded SQL client/server connection is illustrated in the figure. For ODBC applications the ODBC driver manager and ODBC driver stand between the application and the interface library.



Upgrading Embedded SQL applications

You can upgrade a SQL Anywhere Version 5 standalone application to use the Adaptive Server Anywhere Version 6 database server by upgrading the database server and the interface library. You do not need to upgrade the database or the client application itself.

The upgrade procedure uses the **compatibility library.** The compatibility library is a dynamic library (a DLL on PCs, shared library on UNIX) that enables Embedded SQL applications to work with both Version 5 and Version 6 database servers.

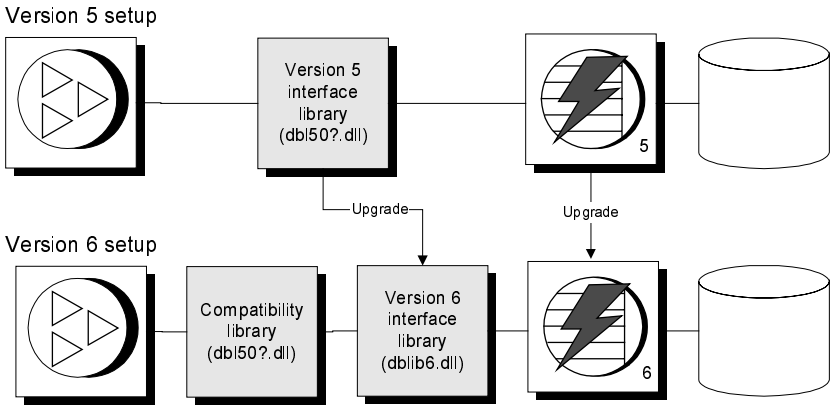
☞ For a description of the compatibility library, see "Using the compatibility library" on page 76.

StartLine connection parameter
If your application connection string uses a StartLine parameter that explicitly provides a *dbclient.exe* or *dbeng50.exe* command line, and this is hard coded into your application, there are additional upgrade considerations.

For information on upgrading applications that use StartLine connection parameters, see "Start parameters and the compatibility library" on page 80.

Components upgraded

The following figure illustrates the changes you need to make in your setup in order to upgrade:



☞ For a description of the compatibility library, see "Using the compatibility library" on page 76.

Upgrade procedures for Embedded SQL applications

This section provides step-by-step procedures for different kinds of Embedded SQL applications:

- Applications using an embedded database are standalone applications using the personal database server (*dbeng50.exe* for Version 5, *dbeng6.exe* for Version 6).
- Client/server applications connect across a network to the network database server. In Version 5, these applications use the *dbclient.exe* executable.

❖ To upgrade Embedded SQL applications that use an embedded database:

1. **Install the Version 6 software** The Version 6 software contains components that enable Version 5 applications to continue working. You can either install into a separate directory or over the top of your Version 5 software.

   The installation places the Version 6 executable directory ahead of the Version 5 executable directory in your system path.

2. **Ensure that your application is using the compatibility DLL** If necessary, copy the compatibility library *dbl50?.dll* from your Adaptive Server Anywhere executable directory to a place where your application will locate it.

   For example, you could copy the compatibility library to the same directory as the module of your application that loads it. The file *dbl50?o.dll* is installed into your Adaptive Server Anywhere executable directory. This directory must be in your path as this library is required by the compatibility library.

   At this stage, your Version 5 application should continue to work as before. However, it will be connecting to your database through the compatibility library rather than directly through the Version 5 interface library.

   ☞ If you have any problems at this stage, you need to check how your application locates the interface library. For information, see "File locations and the compatibility library" on page 79.

3. **Create a new connection description** If your application obtains its connection parameters from configuration files, batch files, or the system registry, you should prepare a new description that uses the Version 6 database server. For example, Sybase Central stores connection descriptions in the system registry.

- If you store a connection string that uses the DBF parameter to start the default database server, then the Version 6 database server is started automatically by the compatibility library, instead of the Version 5 database server. In this case, no new connection description is needed.
- If you store a connection string that uses a Start Line parameter specifying *dbeng50.exe*, you must replace this with one specifying *dbeng6.exe*. If your application contains a hard-coded connection string, you need to take extra steps at this point.

  For more information on upgrading StartLine parameters, see "Start parameters and the compatibility library" on page 80.
- If the database server is started in some other way, such as by a batch file or using a Windows NT service, you must reconfigure this so that the Version 6 database server is started instead.

4 **Use the new connection description** With this step, you are using all Version 6 software and have completed your upgrade. The database itself does not need to be upgraded to work with existing applications.

❖ To upgrade Embedded SQL client/server applications:

1 **Prepare for the upgrade** This step must be carried out at each client machine. You prepare for the upgrade by installing the Version 6 software.

The installation places the Version 6 executable directory ahead of the Version 5 executable directory in your system path.

2 **Ensure that your application is using the compatibility DLL** This step must be carried out at each client machine. If necessary, copy the compatibility library *dbl50?.dll* from your Adaptive Server Anywhere executable directory to a place where your application will locate it.

For example, you could copy the compatibility library to the same directory as the module of your application that loads it. The file *dbl50?o.dll* is installed into your Adaptive Server Anywhere executable directory. This directory must be in your path as this library is required by the compatibility library.

At this stage, your Version 5 application should continue to work as before. However, it will be connecting to your database through the compatibility library rather than directly through the Version 5 interface library.

If you have any problems at this stage, you need to check how your application locates the interface library. For information, see "File locations and the compatibility library" on page 79.

3 **Create a new connection description** This step must be carried out at each client machine. If your application obtains its connection parameters from configuration files, batch files, or the system registry, you should prepare a new description that uses the Version 6 database server. This description is for use when the server is upgraded.

If you store a connection string that uses a StartLine connection parameter specifying *dbclient.exe*, you must replace this with a new one. The new connection description should either contain all the dbclient information as a set of parameters or should specify *dbcli6.exe* instead of *dbclient.exe*. If your application contains a hard-coded connection string, you need to take extra steps at this point.

For more information on upgrading StartLine parameters, see "Start parameters and the compatibility library" on page 80.

For more information on creating connection descriptions that capture the dbclient command-line information, see "Capturing dbclient command-line information" on page 82.

4 **Upgrade the database server** This step must be carried out at the server machine.

- As with any software upgrade, back up your database before upgrading.
- Install Adaptive Server Anywhere on the server machine.
- Start the Version 6 database server on the database.

5 **Use the new connection description** This step must be carried out at each client machine. You need to use the new connection description to connect to the Version 6 server. With this step, you are using all Version 6 software and have completed your upgrade. The database itself does not need to be upgraded to work with existing applications.

Using the version 5 utilities with Adaptive Server Anywhere 6.0

For the Version 5 database utilities, connection strings are supplied interactively. The Version 5 database utilities such as ISQL are Embedded SQL applications that search for the interface library in the following order:

1 The current directory

2 The Version 5 executable directory

3 The system path

For these applications, even though the compatibility library is ahead of the Version 5 interface library in the system path, the Version 5 compatibility library is located.

❖ **To use Version 5 utilities with Adaptive Server Anywhere 6.0:**

1. Make a backup copy of your interface library file.
2. Copy the compatibility library from your Version 6 executable directory to your Version 5 directory. For example, on Windows 95 and Windows NT, copy the file *dbl50t.dll* from the *win32* subdirectory of your Version 6 installation to the *win32* subdirectory of your Version 5 installation.
3. You can now run your Version 5 utilities against both Version 5 and Version 6 database servers.

Upgrading ODBC applications

You can upgrade Version 5 ODBC applications in the following ways:

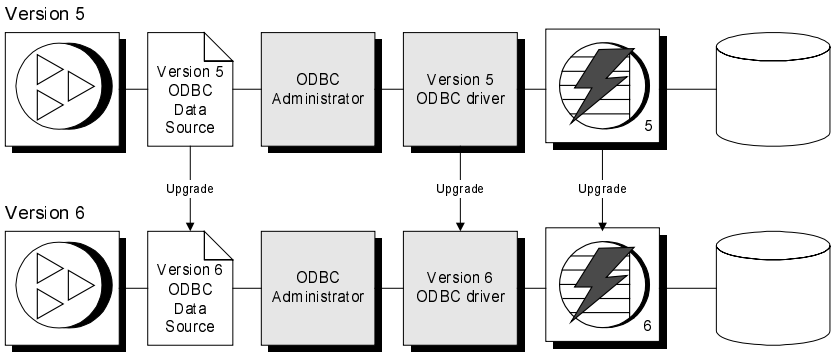
- Replace the Version 5 ODBC data source with a Version 6 ODBC data source. This approach is a complete upgrade, and is described in this section.
- Use the compatibility library to connect to a Version 6 database server. If you choose this route, your application continues to use the Version 5 ODBC driver, so this is not a complete upgrade. The procedure for upgrading in this way is the same as for Embedded SQL applications.

  For information about upgrading in this manner, see "Upgrading Embedded SQL applications" on page 66.

StartLine parameter
If your ODBC data source contains a start line specifying the Version 5 standalone engine (*dbeng50.exe*) or the SQL Anywhere client (*dbclient.exe*), you cannot upgrade using the compatibility library.

Components upgraded

The following figure illustrates the changes you must make when upgrading an ODBC standalone application.



Data sources

The ODBC data source specifies which ODBC driver to use. When an ODBC data source is created as an Adaptive Server Anywhere Version 6 data source, it uses the Version 6 ODBC driver.

Data source must be Version 6
ODBC applications require a Version 6 ODBC data source to work with a Version 6 ODBC driver.

Upgrade procedures for ODBC applications

This section provides step-by-step procedures for different kinds of ODBC applications:

- Applications using an embedded database are standalone applications using the personal database server (*dbeng50.exe* for Version 5, *dbeng6.exe* for Version 6).
- Client/server applications connect across a network to the network database server. In Version 5, these applications use the *dbclient.exe* executable.

Some applications allow you to change the ODBC data source name you use. Other applications use a fixed data source name. You can upgrade either kind of application.

❖ To upgrade ODBC applications that use an embedded database:

1. **Install the Version 6 software** The Version 6 software contains components that enable Version 5 ODBC applications to continue working. You can either install into a separate directory or over the top of your Version 5 software.

   Your application should be unaffected by installation of the Version 6 software.

2. **Create a Version 6 ODBC data source** The changes you make depend on the connection parameters you use in the data source.

   - If you start the default database server using the DBF parameter, you can use the same connection parameters in your new data source as your old one.
   - If you store a connection string that uses a Start Line parameter specifying *dbeng50.exe*, you must replace this with one specifying *dbeng6.exe*.

     For more information on upgrading StartLine parameters, see "Start parameters and the compatibility library" on page 80.
   - If the database server is started in some other way, such as by a batch file or using a Windows NT service, you must reconfigure this so that the Version 6 database server is started instead.

   For information on creating Version 6 data sources, see "Working with ODBC data sources" on page 42 of the book *Adaptive Server Anywhere User's Guide*.

3 **Use the new data source** With this step, you are using all Version 6 software and have completed your upgrade. The database itself does not need to be upgraded to work with existing applications.

Some applications may have the data source name hard wired. In this case, you need to replace the Version 5 data source with a Version 6 data source of the same name. It is recommended that you rename, rather than delete, your Version 5 data source.

❖ **To upgrade ODBC client/server applications:**

1 **Prepare for the upgrade** This step must be carried out at each client machine. You prepare for the upgrade by installing the Version 6 software.

Your application should be unaffected by installation of the Version 6 software.

2 **Create a Version 6 ODBC data source** This step must be carried out at each client machine. The changes you make depend on the connection parameters you use in the data source.

- If you start the default database server using the DBF parameter, you can use the same connection parameters in your new data source as your old one.
- If you store a connection string that uses a StartLine connection parameter specifying *dbclient.exe*, you must replace this with a new one. The new connection description should either contain all the *dbclient* information as a set of parameters or should specify *dbcli6.exe* instead of *dbclient.exe*. If your application contains a hard-coded connection string, you need to take extra steps at this point.

  For more information on upgrading StartLine parameters, see "Start parameters and the compatibility library" on page 80.

  For more information on creating connection descriptions that capture the dbclient command-line information, see "Capturing dbclient command-line information" on page 82.
- If the database server is started in some other way, such as by a batch file or using a Windows NT service, you must reconfigure this so that the Version 6 database server is started instead.

For information on creating Version 6 data sources, see "Working with ODBC data sources" on page 42 of the book *Adaptive Server Anywhere User's Guide*.

3 **Upgrade the database server** This step must be carried out at the server machine.

- As with any software upgrade, back up your database before upgrading.
- Install Adaptive Server Anywhere on the server machine.
- Start the Version 6 database server on the database.

4 **Use the new data source** This step must be carried out at each client machine. You need to use the new connection description to connect to the Version 6 server. With this step, you are using all Version 6 software and have completed your upgrade. The database itself does not need to be upgraded to work with existing applications.

Upgrade notes for PowerBuilder and InfoMaker users

Users of Sybase PowerBuilder and InfoMaker should make some changes in order to obtain full functionality with Adaptive Server Anywhere Version 6.

The pbodb60.ini file

PowerBuilder and InfoMaker use a file named *pbodb60.ini* to hold ODBC data source information. The 60 in the file name may be different, depending on the version you have. For each ODBC driver it provides such things as DDL syntax, default DBParm options, valid function names and special datatypes.

If your *pbodb60.ini* file does not have a Adaptive Server Anywhere section, PowerBuilder and InfoMaker default to a core syntax. This limits the operations you can carry out using these tools. For example you cannot create, alter, or drop primary and foreign keys.

Upgrading your pbodb60.ini file

To obtain complete functionality with PowerBuilder and InfoMaker, you need to upgrade your *pbodb60.ini* file.

❖ **To upgrade your pbodb60.ini file:**

1 Make a backup copy of your existing pbodb60.ini file.

2 Add an Adaptive Server Anywhere section to the working copy of the file containing the same information as the existing Sybase SQL Anywhere section:

```
[Adaptive Server Anywhere]
PBSyntax='WATCOM50_SYNTAX'
PBDateTime='STANDARD_DATETIME'
PBFunctions='WATCOM_FUNCTIONS'
PBDefaultValues='autoincrement,current date,current
time,current timestamp,timestamp,null,user'
PBDefaultCreate='YES'
PBDefaultAlter='YES'
PBDefaultExpressions='YES'
```

```
DelimitIdentifier='YES'
PBDateTimeInvalidInSearch='NO'
PBTimeInvalidInSearch='YES'
PBQualifierIsOwner='NO'
PBSpecialDataTypes='WATCOM_SPECIALDATATYPES'
IdentifierQuoteChar='"'
PBSystemOwner='sys,dbo,rs_systabgroup'
PBUseProcOwner='YES'
SQLSrvrTSName='YES'
SQLSrvrTSQuote='YES'
SQLSrvrTSDelimit='YES'
ForeignKeyDeleteRule='Disallow if Dependent Rows
Exist (RESTRICT),Delete any Dependent Rows
(CASCADE),Set Dependent Columns to NULL (SET NULL)'
TableListType='GLOBAL TEMPORARY'
```

Using the compatibility library

The **compatibility library** is a dynamic library (a DLL on PCs, shared library on UNIX) that enables Embedded SQL applications to work with both Version 5 and Version 6 database servers. This section describes how the compatibility library works.

Who needs to read this section?
You should read this section if you are upgrading SQL Anywhere Version 5 to Adaptive Server Anywhere Version 6, and have existing Embedded SQL applications that you need to work with the Version 6 server.

The Version 5 Embedded SQL interface library

All client machines running SQL Anywhere Version 5 applications, whether connecting over a network or to a personal server, have a SQL Anywhere Version 5 **interface library**.

- For Windows 95 and Windows NT, this library is a DLL named *dbl50t.dll*.
- For Windows 3.x, this library is a DLL named *dbl50w.dll*.
- For OS/2, this library is a DLL named *dbl502.dll*.

In this documentation, these files are written as *dbl50?.dll*.

How Version 5 client applications locate the interface library

Version 5 client applications locate the interface library in one of the following ways:

- **ODBC applications** ODBC applications connect to a SQL Anywhere database using the SQL Anywhere ODBC driver. The SQL Anywhere Version 5 ODBC driver calls functions in the Version 5 Embedded SQL interface library.
- **Embedded SQL applications** Embedded SQL applications for Windows operating systems and OS/2 call into the interface library. In these calls, the interface library is referenced by name—*dbl50t.dll* for Windows 95 and NT.

  The SQL Anywhere Version 5 ODBC driver is an Embedded SQL application.

Using the compatibility library

The **compatibility library** is *optionally* installed as part of the Version 6 client software. It provides support for two interface libraries at the same time. You should check your installation to confirm that it is installed.

The setup program should ensure that your application calls the compatibility library instead of the Version 5 interface library.

The setup program carries out the following steps to ensure that applications call the interface library.

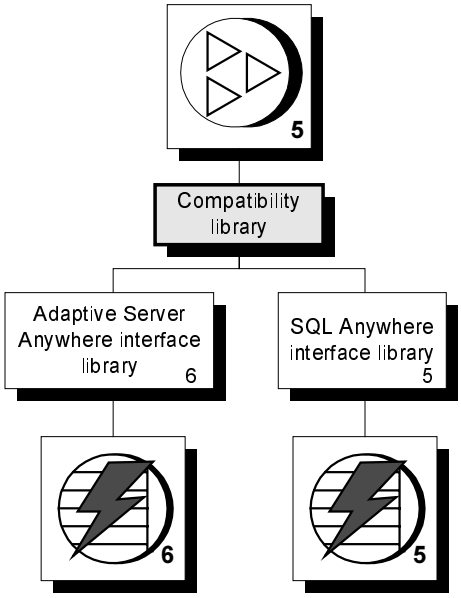
- The compatibility library has the same name as the SQL Anywhere Version 5 interface library. For example, on Windows NT, the compatibility library is named *dbl50t.dll*.
- The compatibility library is installed into the same directory as other Version 6 software.
- The Version 6 installation directory is placed ahead of the Version 5 directory in the system path. This ensures that applications locate the compatibility library ahead of the Version 5 interface library.
- A Version 5 interface library is installed into the same directory as the compatibility library, but with the name *dbl50to.dll*. When the compatibility library is accessing Version 5 servers, it calls this interface library.
- The Version 6 interface library is installed into the same directory as the compatibility library. It has the name *dblib6.dll* on Windows NT. When the compatibility library is accessing Version 6 servers, it calls this interface library.

If you have problems using the compatibility library, you should check the order of the directories in your path, and ensure that the Version 6 location is ahead of the Version 5 location in the path.

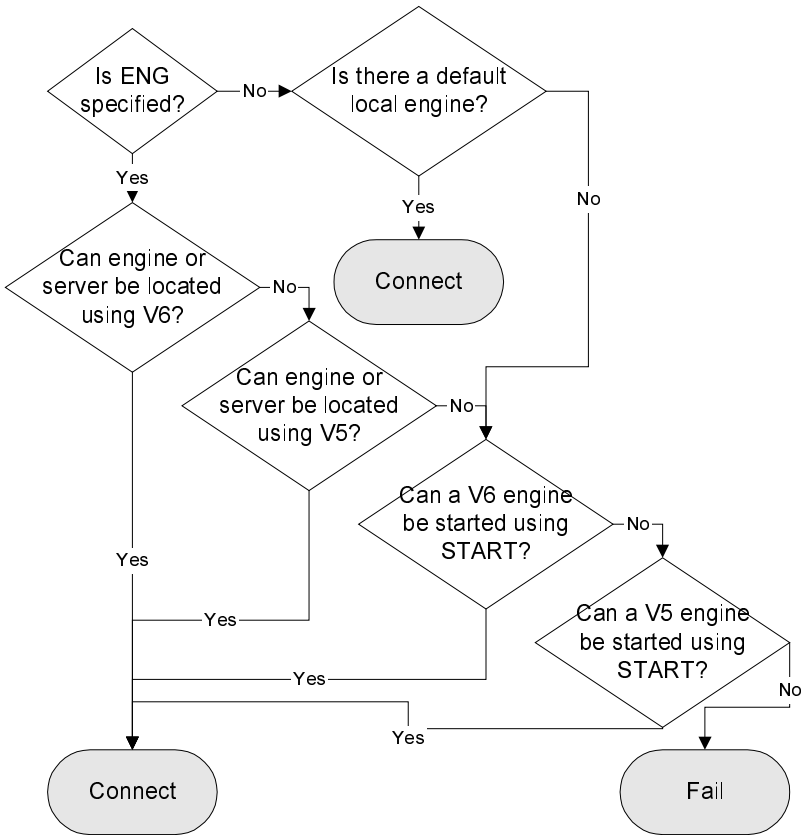
How the compatibility library works

Using the supplied connection string, the compatibility library attempts to connect to an Adaptive Server Anywhere database using the Adaptive Server Anywhere Version 6 interface library. If this attempt fails, it attempts to connect to a SQL Anywhere database using the SQL Anywhere Version 5 library.

The following figure illustrates how the compatibility library enables communications to both a SQL Anywhere Version 5 and an Adaptive Server Anywhere Version 6 database server. The number in the lower right hand corner of the boxes indicates the version of the software component.



The following figure illustrates the algorithm used by the compatibility library to connect to a server:



File locations and the compatibility library

The compatibility library and the Version 5 interface library have the same file name (*dbl50?.dll*). For your application to use the compatibility library, it must locate it ahead of the Version 5 interface library when it searches for DLLs.

To ensure that your application locates the compatibility library ahead of the Version 5 interface library you must understand how your application searches for DLLs.

Searching for DLLs

The Version 6 installation program places the Version 6 executable directory ahead of the Version 5 directory in the system path, so any application that is searching in the system path for *dbl50?.dll* will find the compatibility library ahead of the Version 5 interface library. If your application searches in a different manner, you must ensure that the compatibility library is located.

Testing to see which library is located

You can test to see which library is located in the following ways:

- Attempt to connect to a Version 6 database server. You cannot connect using the Version 5 interface library.
- Specify a database file parameter (DBF) and no start line in your connection string. If the Version 5 interface library is located, the SQL Anywhere Version 5 standalone database engine is started. If the compatibility library is located, the Version 6 personal database server is started.

Start parameters and the compatibility library

Applications using a connection string that includes a StartLine connection parameter face some additional issues in upgrading.

The StartLine parameter provides explicit instructions for starting the database engine or the SQL Anywhere Client executable. Sample StartLine parameters are as follows:

- **Standalone application** A sample StartLine parameter for a Version 5 standalone application is as follows:

  ```
  dbeng50.exe -c 8M
  ```

- **Network client application** A sample StartLine parameter for a Version 5 network client application is as follows:

  ```
  dbclient.exe -x tcpip
  ```

In Version 6, the *dbeng50.exe* executable is replaced by the personal database server *dbeng6.exe*. The dbclient.exe executable is no longer required in Version 6, but a compatibility client (*dbcli6.exe*) is provided.

For a description of the compatibility client, see "Capturing dbclient command-line information" on page 82.

❖ **To upgrade StartLine parameters:**

- The procedure depends on where your connection parameters are stored.
  - If your connection parameters are stored outside the application itself, then you need to alter the connection parameters to use the appropriate *dbeng6.exe* or *dbcli6.exe* executable names instead of *dbeng50.exe* and *dbclient.exe*.

All the information on the dbclient command line can be rephrased in terms of other connection parameters. You can avoid using the dbcli6.exe compatibility client by rewriting your connection parameters using these parameter. For information, see "Connection and Communication Parameters" on page 39 of the book *Adaptive Server Anywhere Reference Manual*.

- If your connection parameters are hard-wired into your application, you must either relink your application with a new connection string, or rename the *dbcli6.exe* compatibility client as *dbclient.exe*, and replace the Version 5 client executable with the renamed Version 6 one.

  There are many possible configurations of client command lines and connection parameters. Be sure you test any solution thoroughly before deploying.

Capturing dbclient command-line information

Version 5 client applications that connect to a database server do so via the SQL Anywhere Client, an executable named *dbclient.exe*. The client executable command line contains information needed to locate a server, including the following:

- **Default server name** The server name on the client command line is the default server name. When a client executable is running, the application does not need to supply a server name in order to connect to the default server.
- **Network communications parameters** A listing of network protocols to use together with a set of communications parameters specifies where the client executable is to look as it attempts to locate a server.
- **Client/Server communication tuning** A set of parameters allows the packet size, buffer size, and so on to be tuned for optimum performance.

In Version 6, this information is held in an ODBC data source along with other connection information. As there is no longer a client executable, there is no longer a client command line. In Version 6, Embedded SQL applications can use ODBC data sources as a source of connection parameters.

Client command line scope

Only one Version 5 client executable can be run at a time, and it may be used by more than one application and handle connections to more than one server. The command-line information is therefore global to the machine.

How to capture client command-line information

During upgrade to Version 6, you must ensure that Version 5 *dbclient* command-line information is captured in such a way that the Version 6 ODBC or Embedded SQL applications can use it. You can do this in one of the following ways:

- **Place the information in an ODBC data source** If the information can be placed in a data source, you can use it with Version 6 ODBC or Embedded SQL applications.
- **Use the SQLCONNECT environment variable** The SQLCONNECT environment variable contains a connection string. It is searched early in the process of establishing which connection parameters to use. You may be able to use SQLCONNECT settings to override connection parameters.

- **Use the dbcli6 executable** If the information cannot be placed in a data source, you can use an executable named *dbcli6.exe*, which makes the command-line information available to the Version 6 interface libraries.

The way to capture client command-line information depends on where the information is located.

Where command-line information is located

Your existing command line information may be held in one of the following places.

- **ODBC data source** The ODBC data source contains a START parameter that can hold a client executable command-line.
- **As a connection string** Your application may obtain client information (for example from an initialization file), and supply it in a connection string as the START parameter.
- **A batch file** You may have a batch file that includes a client executable command line as part of your startup process.
- **Under an icon** You may have a client executable command line under an icon on your desktop.

How to capture the information

- **From an ODBC data source** The ODBC data source upgrade captures the information in an ODBC data source START parameter.
- **From a batch file or under an icon** You can replace the reference to *dbclient.exe* in the batch file with *dbcli6.exe*. The *dbcli6.exe* executable is provided for compatibility reasons only, and makes the command-line parameters available to Version 6 interface libraries.
- **From a connection string** If the application retrieves information from an initialization file or other editable source, you can replace references to *dbclient.exe* with *dbcli6.exe*.
- **Hard-wired connection strings** Only if you have a hard-wired connection string in your application (that is, one that cannot be edited), you must alter the source of the application and recompile.

How the compatibility client executable works

The Version 6 compatibility client executable does not serve the same purpose as the Version 5 client executable. Instead, it simply places the command-line parameters into a specially-named data source and terminates.

The compatibility data source is named **dbcli6 Connection Parameters**.

When dbcli6.exe is needed

The Version 6 compatibility client is only needed when the existing Version 5 application runs the client from outside an ODBC data source.

If the ODBC data source contains a START parameter that holds the dbclient command line, this information is captured in Version 6 form when the ODBC data source is upgraded, and you do not need to use the Version 6 compatibility client.

If you store the client command line in a batch file, under an icon on your desktop, or in an initialization file used by an Embedded SQL application, you need to replace *dbclient* with *dbcli6.exe* in this command line.

Typical use for the compatibility client

The most common method of using the client in Version 5 applications is as follows:

1 The *dbclient* command line includes a server name. This server name is the default server for connections made through that client.

2 The client application runs without specifying a server. The Version 5 interface library locates the client, and connects to its default server.

How the compatibility data source works

When you replace *dbclient* with *dbcli6* in the client command line, the following set of steps takes place:

1 When the *dbcli6* command line is executed, the compatibility client creates a data source named **dbcli6 Connection Parameters**, and then stops.

The special data source, named **dbcli6 Connection Parameters**, is called a **compatibility data source**. The name is reserved for use by the compatibility client, and you should not use it for any other purpose.

2 The client application starts. Its connection string is handed to the compatibility library.

3 The compatibility library attempts to connect to a Version 6 server.

- The connection string does not contain a server name (no ENG connection parameter). The next step in the process is to search for a data source.
- As the connection string is from a Version 5 application, it has no DSN or FileDSN parameter (these parameters were not available for dbclient in Version 5). The next step is to look for a compatibility data source.
- The **dbcli6 Connection Parameters** data source is located, and the server name and other parameters from the data source used to enable a connection to the server.

4 When the application disconnects from the server, the Version 6 interface library deletes the **dbcli6 Connection Parameters** data source from the list of available data sources.

This behavior allows other applications to use the compatibility data source to establish connections while the first application is running, and prevents it from being used once that application has closed down (although established connections are not affected). The lifetime of the compatibility data source is thus similar to that of the Version 5 client.

The compatibility data source is only used if no other data source is specified in the connection string or in the SQLCONNECT environment variable.

Upgrading databases

If you wish to use the new features of Adaptive Server Anywhere Version 6, you need to upgrade your database to Adaptive Server Anywhere 6. The format of the database files is the same as in SQL Anywhere Version 5, but the system tables and available database options have changed.

> ***Caution***
> *Ensure that you back up your database before upgrading.*

Supported versions

You can upgrade your database from any of the following versions of the software to Adaptive Server Anywhere Version 6 format:

- **SQL Anywhere Version 5** Including versions 5.0 and 5.5, all patch levels.
- **Watcom SQL Version 4** All patch levels.
- **Watcom SQL Version 3.2** Only patch level **e** and above.

What the upgrade utility does

The upgrade utility carries out the following tasks:

- **Adds new system tables** For example, the SYSJAR and SYSJAVACLASS system tables are new in Version 6, and must be present for any of the Java capabilities of Adaptive Server Anywhere to be available.
- **Adds new columns to existing tables** For example, with the introduction of user-defined data types in SQL Anywhere Version 5, two system tables had a **user_type** column added.
- **Adds database options** Database options provide a degree of tuning for overall database behavior or for individual user behavior. The list of available options is kept in the system tables, and needs to be upgraded for the options to become available.

❖ To upgrade a database to Adaptive Server Anywhere Version 6 using Sybase Central:

1. Make a backup of your database using the backup utility.
2. Upgrade the database using the upgrade utility.

❖ To upgrade a database to Adaptive Server Anywhere Version 6 using command line utilities:

1. Make a backup of your database using the backup utility.
2. Upgrade the database using the upgrade utility.

Upgrading SQL Remote installations

SQL Remote installations include a consolidated database and many remote databases, together with a Message Agent at each site.

At each site, the Message Agent handles the sending and receiving of messages. The messages take the form of SQL statements, and the database server handles the actual execution of those SQL statements.

The upgrade requirements for SQL Remote are as follows:

- **No need to upgrade databases** There is no need to upgrade databases for Version 6. However, if you do wish to upgrade a database, you can do so using the Upgrade utility at any of the sites. There is no need for all databases to be upgraded at the same time.
- **Software upgrades can be one site at a time** Version 5 Message Agents can exchange messages with Version 6 Message Agents as long as the COMPRESSION database option is set to a value of -1 (minus one). There is no need to upgrade software throughout the installation simultaneously.
- **Message Agent and server can be upgraded separately** The Message Agent is an Embedded SQL application. Therefore, the database server can be upgraded before the Message Agent as long as the compatibility library is used. It is, however, recommended that you upgrade your Message Agent at the same time as the database server for performance reasons.

  The Message Agent cannot be upgraded before the database server, as a Version 6 client application cannot work with a Version 5 server.
- **Do not unload and reload** There is no need to unload and reload a database when moving from Version 5 to Version 6.

  Replication is based on the transaction log, and when a database is unloaded and reloaded, the old transaction log is no longer available. For this reason, good backup practices are especially important when participating in replication.

Example

One approach to upgrading is as follows:

1. Upgrade the consolidated database server and Message Agent. Set the COMPRESSION database option to -1 so that all messages are compatible with the Version 5 software at remote sites.
2. Over time, upgrade remote database servers and Message Agents. You can set the COMPRESSION database option to a value other than -1 to take advantage of compression and encoding on messages being sent to the consolidated database server.

3 When all remote database servers and Message Agents are upgraded, set the COMPRESSION database option a the consolidated site to a value other than -1.